SoftwareX 27 (2024) 101783

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original Software Publication

An application for solving minimization problems using the Harmony search algorithm

Fazliddin Makhmudov, Dusmurod Kilichev, Young Im Cho *

Department of Computer Engineering, Gachon University, Seongnam, South Korea

A R T I C L E I N F O

Keywords:
Harmony search algorithm
Minimization problem

A B S T R A C T

The Harmony Search (HS) algorithm, inspired by the improvisational process of musicians, offers a novel and effective approach to optimization problems. This paper presents an application designed to solve a wide range of minimization problems using the HS algorithm. By simulating the creative exploration and refinement seen in musical harmonies, the HS algorithm efficiently navigates complex solution landscapes, delivering high accuracy and computational efficiency. Illustrated through various optimization examples, this tool showcases the versatility and power of the HS algorithm in addressing linear, non-linear, and discrete models. Our work highlights the practical utility of bio-inspired algorithms in solving real-world problems, providing a user-friendly platform for researchers and engineers to harness the potential of the HS algorithm in diverse fields.

Metadata

Nr	Code metadata	
C1	Current code version	v01.00
C2	Permanent link to code/ repository used for this code version	https://github.com/TATU-hacker/Solver_of_minimization_problems.git
C3	Legal code license	GNU General Public License v3.0
C4	Code versioning system used	none
C5	Software code languages, tools and services used	Python PyQt5 - version 5.15.0 Numpy - version 1.19.1 Matplotlib - version 3.3.1
C6	Support email for questions	dusmurod@gachon.ac.kr

1. Motivation and significance

The pursuit of optimal solutions is a fundamental aspect of scientific and engineering research, driving the development of algorithms designed to navigate complex problem spaces. Within this context, the Harmony Search (HS) algorithm stands out as a distinguished approach, inspired by the improvisation process of musicians seeking harmonious melodies [1]. This innovative algorithm, particularly effective in addressing minimization problems, demonstrates a compelling narrative of adaptability, efficiency, and broad applicability, which forms the foundation of our application. Our motivation for leveraging the HS algorithm stems from its unique ability to mimic the stochastic and creative exploration of solutions, making it exceptionally well-suited for tackling the multifaceted challenges inherent in minimization problems.

Minimization problems, characterized by their vast solution landscapes and stringent constraints, are ubiquitous across numerous disciplines, including logistics, finance, engineering, and information technology. Traditional methods often struggle with the limitations imposed by the complexity, non-linearity, and multi-dimensionality of these problems. In this context, our application, powered by the HS algorithm, emerges as a leading solution, embodying a blend of simplicity and sophistication. By leveraging the metaphorical process of seeking harmonious solutions, our application transcends conventional optimization boundaries, enabling the efficient identification of near-optimal solutions with remarkable computational efficiency [2-8].

The significance of this application extends beyond the technical realm, offering a versatile tool that democratizes access to advanced optimization capabilities. For practitioners and researchers alike, it opens new avenues for exploring solution spaces with an intuitive understanding, mirroring the improvisational search for harmony. This approach not only enhances the problem-solving toolkit available to industries and academia but also fosters a deeper appreciation for the elegance inherent in algorithmic design inspired by natural processes.

* Corresponding author.
E-mail address: yicho@gachon.ac.kr (Y.I. Cho).

https://doi.org/10.1016/j.softx.2024.101783
Received 7 April 2024; Received in revised form 31 May 2024; Accepted 3 June 2024

In essence, our application represents a significant advancement in the field of optimization, bridging theoretical innovation with practical utility. The use of the HS algorithm in tackling minimization problems epitomizes the harmonious balance between creativity and analytical rigor, offering a beacon for future explorations in optimization. Through this endeavor, we underscore the pivotal role of interdisciplinary inspiration in algorithm development, promising a future where the most challenging problems can be approached with a blend of intuition and precision, much like the musicians whose quest for harmony inspires our algorithmic journey.

The HS algorithm exemplifies bio-inspired optimization techniques, drawing its conceptual framework from the musical improvisation process observed in jazz musicians. This innovative approach to solving optimization problems, particularly those of a minimization nature, encapsulates the essence of creativity and adaptability. Below, we succinctly delineate the core mechanism underpinning the HS algorithm, presenting it through a refined academic lens [1].

The HS algorithm begins its optimization journey by initializing the Harmony Memory (HM) with diverse, randomly generated solution vectors. This initial collection forms the foundation for the algorithm's iterative refinement and exploration, much like a composer starting with a variety of musical themes. The algorithm's success hinges on three key parameters [9]. The first, Harmony Memory Size (HMS), determines the capacity of the HM, influencing the diversity and depth of explored solutions. A larger HMS allows for a broader spectrum of potential solutions. The second parameter, Harmony Memory Considering Rate (HMCR), balances the algorithm's use of existing harmonies versus generating new ones. This balance ensures a thorough search by harmonizing exploitation of known solutions with exploration of new possibilities, akin to a composer alternating between familiar motifs and new melodies. The third parameter, Pitch Adjusting Rate (PAR), allows for the fine-tuning of solutions, much like a musician's subtle adjustments to achieve perfect pitch. This parameter enables precise modifications to enhance solution quality.

As the algorithm progresses, it enters the Harmony Generation phase, where new harmonies are crafted either by selecting from the HM or generating new values, with the potential for pitch adjustments. This phase resembles an improvisational performance, combining familiar and spontaneous elements. Next, in the Memory Update phase, the algorithm evaluates and integrates superior harmonies into the HM, discarding less optimal ones. This continuous refinement process enhances the overall solution quality, much like a composer refining their composition.

The journey culminates in the Optimal Solution Identification phase. The algorithm iterates through harmony generation and memory update until a stopping criterion, such as a maximum number of iterations, is met. The finest harmony in the HM is then identified as the optimal solution, akin to a perfected symphony ready for performance. Through these phases, the Harmony Search algorithm exemplifies a sophisticated approach to optimization, balancing exploration and exploitation to uncover optimal solutions. This process mirrors the artistic journey of composing and perfecting a masterpiece.

In essence, the elegance of the HS algorithm lies in its mimicry of the creative and iterative process of musical improvisation, adeptly adapted to the domain of optimization. This methodology not only enriches the computational optimization landscape but also underscores the profound potential of interdisciplinary inspiration in algorithmic design, offering a harmonious blend of exploration, exploitation, and solution refinement.

2. Software description

The software tool was developed using the Python programming language. PyQt5 was utilized to create the graphical user interface (GUI) of the application. For interactive visualizations, the application employed Matplotlib, a comprehensive plotting library. Additionally, the NumPy library was used to perform array operations, enhancing the computational efficiency and functionality of the tool.

The software interface is modern, simple, and intuitive, ensuring ease and convenience. As shown in Fig. 1, the $f(x)$ field allows users to input their desired functions directly. All entered functions are collected in a functions.txt file, where necessary functions can be pre-included for convenience. The 'Select a predefined function' button opens a window to choose from predefined functions, as depicted in Fig. 2. This design makes managing and utilizing mathematical functions both efficient and user-friendly.

In the subsequent step, users are prompted to enter the parameters of the function. They begin by specifying the number of iterations, which defines how many times the algorithm will run. Next, they set the HMS, which determines the size of the harmony memory. Users then define the intervals for the evolution of the memory considering rate with $\mathrm{HMCR}_{\min}$ and $\mathrm{HMCR}_{\max}$. Similarly, the intervals for the pitch adjusting rate are set using $\mathrm{PAR}_{\min}$ and $\mathrm{PAR}_{\max}$. Finally, the range for the harmony search bandwidth is defined with $\mathrm{bw}_{\min}$ and $\mathrm{bw}_{\max}$.

Should any parameter be entered incorrectly, an appropriate message immediately appears, blocking further progression to prevent errors. Once the parameters are correctly entered, pressing the 'Next' button brings up a window for retrieving the ranges of values for the variables detected in the equation, as depicted in Fig. 3. Again, if any parameter is incorrect, a message will display, ensuring that users can only proceed with correct and valid inputs.

Upon successful data input and the calculation of the solution, the main window displays a function contour plot along with the best points identified by the algorithm (Fig. 4). This visual representation allows users to see the optimal solutions in the context of the function's landscape. Additionally, the solution field shows the values of the variables and the function's value at the identified optimal point, providing a clear and concise summary of the results.

The chart includes a toolbar from the Matplotlib package, enabling users to perform basic operations such as zooming and saving the chart to a file. This feature enhances the user experience by allowing for easy manipulation and preservation of the visual data.

This seamless and user-friendly interface design ensures that managing and optimizing functions within the application is both efficient and error-free, providing clear visual feedback and easy-to-use tools for analysis and presentation.

3. Illustrative examples

We consider the following function:

$$f(\overrightarrow{x})=\left\{1+(x_1+x_2+1)^2\times\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right\}$$
$$\times\left\{3+(2x_1-3x_2)^2\times\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right\}.$$

$$\min f(\overrightarrow{x})=f(0,\ 1)=3.$$

For this function, an algorithm was run with various combinations of parameter values. These values for each test case are shown in the Table 1 below. The ranges of the variables are:

$$\begin{cases}-2<x_1<2,\\ -2<x_2<2.\end{cases}$$

In our exploration of the HS algorithm's performance, we encountered a variety of outcomes based on the different parameter settings applied. Initially, numerous calls to the first case with 100 iterations revealed significant variability in results, indicating that this number of iterations was insufficient for reliable conclusions (Case 1).

When the number of iterations was increased to 1000, the program more consistently returned a minimum value close to the global minimum, demonstrating a marked improvement in accuracy (Case 2). However, further adjustments revealed additional complexities.

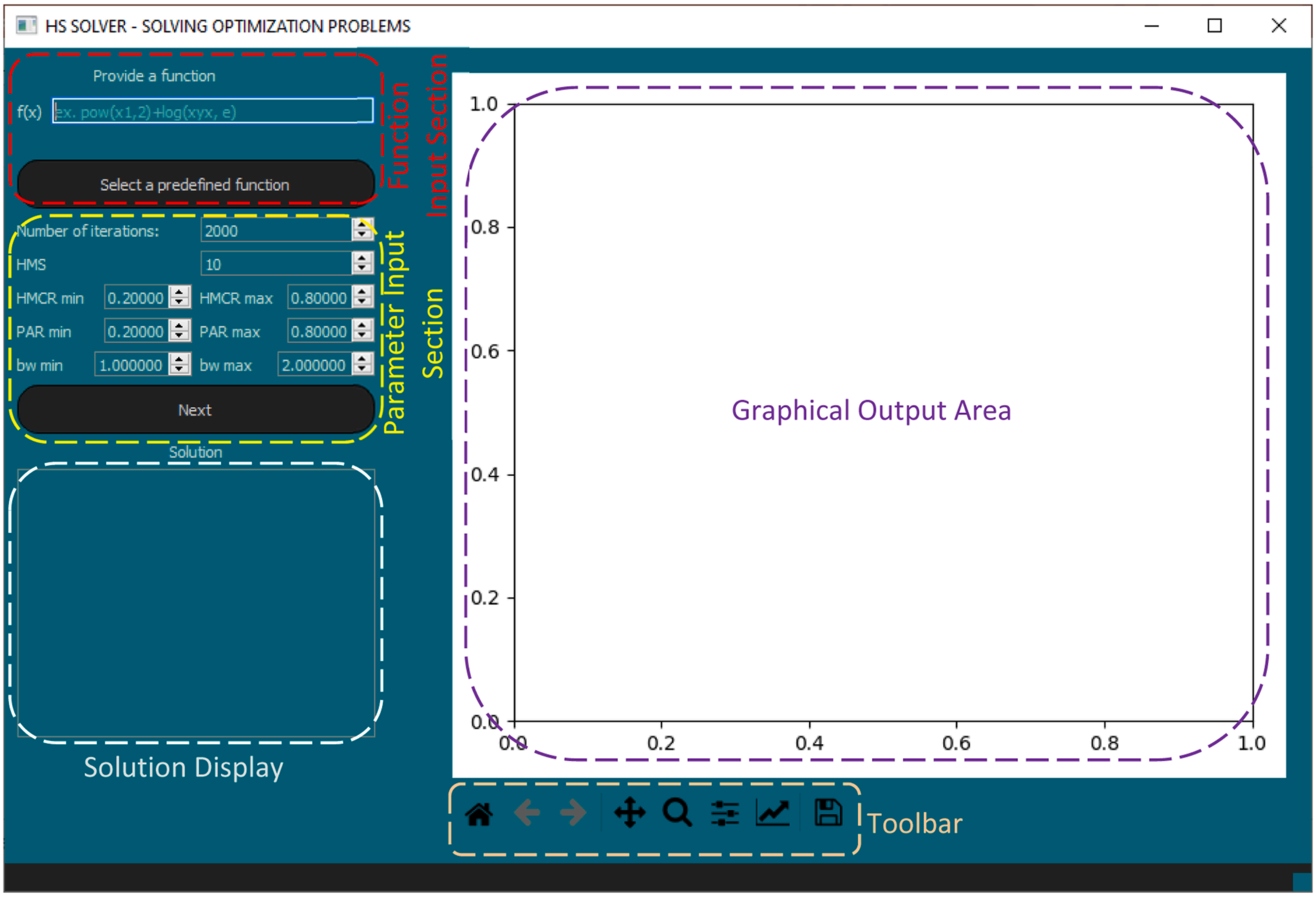

Fig. 1. Software interface.

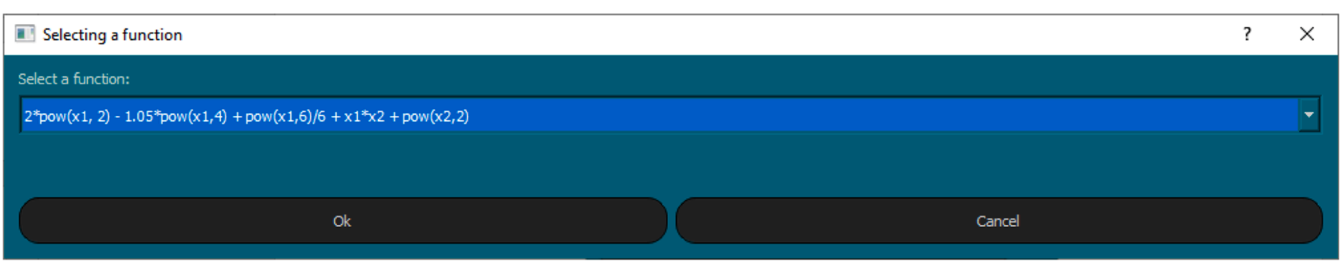

Fig. 2. Select functions.

Increasing the parameter HMS in Case 3 resulted in the algorithm finding fewer new vectors and yielding less accurate results. This suggested that a larger HMS might reduce the algorithm's effectiveness by limiting the exploration of new solutions.

A particularly interesting finding emerged when setting HMCR to 0 in Case 4. Here, we observed that changes to other parameters had no noticeable impact on the algorithm's operation. The results were imprecise because, with HMCR set to 0, the parameters PAR and bw were rendered irrelevant; the algorithm bypassed the stages where these parameters would typically be utilized. Additionally, HMS was inconsequential since new points were selected randomly from the entire search range, not influenced by stored harmonies.

Conversely, setting HMCR to 1 in Case 5 revealed that the algorithm struggled to find any solution with a low HMS. However, when HMS was increased, the algorithm began to perform better, identifying values close to the optimal (Case 6). This improvement can be attributed to the fact that with HMCR at 1, all new points are generated based on stored points, allowing the algorithm to search for new coordinates within the bandwidth (bw) from each stored coordinate. Therefore, a larger HMS enhances the algorithm's capacity to explore more extensively within the domain.

In Case 8, with HMCR set to 1 and PAR set to 0, the algorithm consistently found only single points in each call. This outcome was linked to the algorithm being constrained to seek the most optimal solution only from the initially drawn HMS points, limiting its exploratory capability.

Setting PAR to 1 in Case 9 did not produce significant changes in the results, suggesting that PAR alone does not dramatically influence the search efficacy. However, when HMCR evolved between 0.2 and 0.8 in Case 10, we noted an improvement over the fixed HMCR of 0. This was because new coordinates were drawn not only from the entire domain but also from the best-remembered solutions, enhancing the search process.

Case 11 highlighted the benefits of pitch adjustment (PAR $>$ 0),

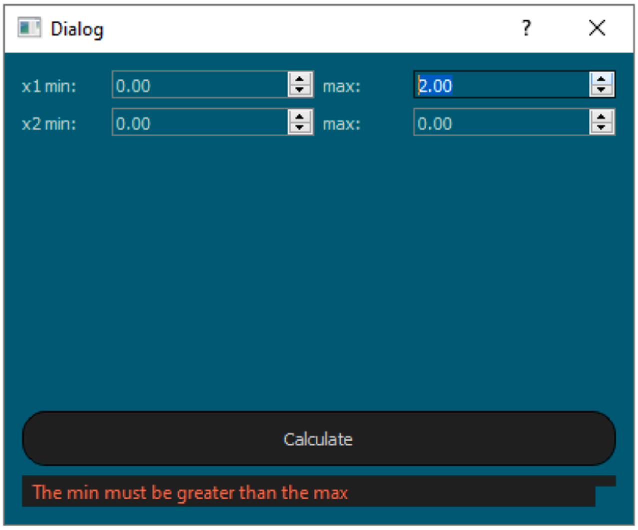

Fig. 3. The range of values of variables.

where more accurate results were obtained compared to scenarios without pitch adjustment. Even better results were achieved when PAR was set within the range of 0.2 to 0.8, as demonstrated in Case 12. This range allowed for a balanced and effective adjustment process, improving solution accuracy.

Finally, Case 13 provided insights into the effects of increasing the bandwidth parameter (bw). While the algorithm traversed all local minima, it eventually identified the global minimum, albeit with slightly less accuracy than when bw had a lower value. This trade-off between thorough exploration and precision is depicted in Fig. 5.

Through these cases, we comprehensively analyzed how various parameter settings influence the HS algorithm's performance, revealing the intricate balance required to optimize its efficiency and accuracy.

4. Impact

The HS algorithm has emerged as a viable alternative to traditional optimization methods, particularly in the field of engineering science. Its application spans various complex optimization problems across diverse engineering domains, making it a widely adopted tool in sectors such as heat exchanger design, steel manufacturing, electronics, mechanical

Table 1
Values for each test case.

ID	L	HMS	HMCR	PAR	bw	Results
1	100	5	0.2–0.8	0.2–0.8	0–1	69.5196
2	1000	5	0.2–0.8	0.2–0.8	0–1	3.4156
3	1000	100	0.2–0.8	0.2–0.8	0–1	5.9362
4	1000	100	0–0	0.2–0.8	0–1	4.8874
5	1000	5	0–0	0.2–0.8	0–1	168.451
6	1000	5	1–1	0.2–0.8	0–1	3.1735
7	1000	100	1–1	0.2–0.8	0–1	263.993
8	1000	10	1–1	0–0	0–1	3.679
9	1000	10	1–1	1–1	0–1	3.478
10	1000	10	0.2–0.8	0–0	0–1	3.052
11	1000	10	0.2–0.8	1–1	0–1	3.1479
12	10,000	10	0.2–0.8	0.2–0.8	0–1	3.045
13	**10,000**	**10**	**0.2–0.8**	**0.2–0.8**	**1–2**	**3.0008**

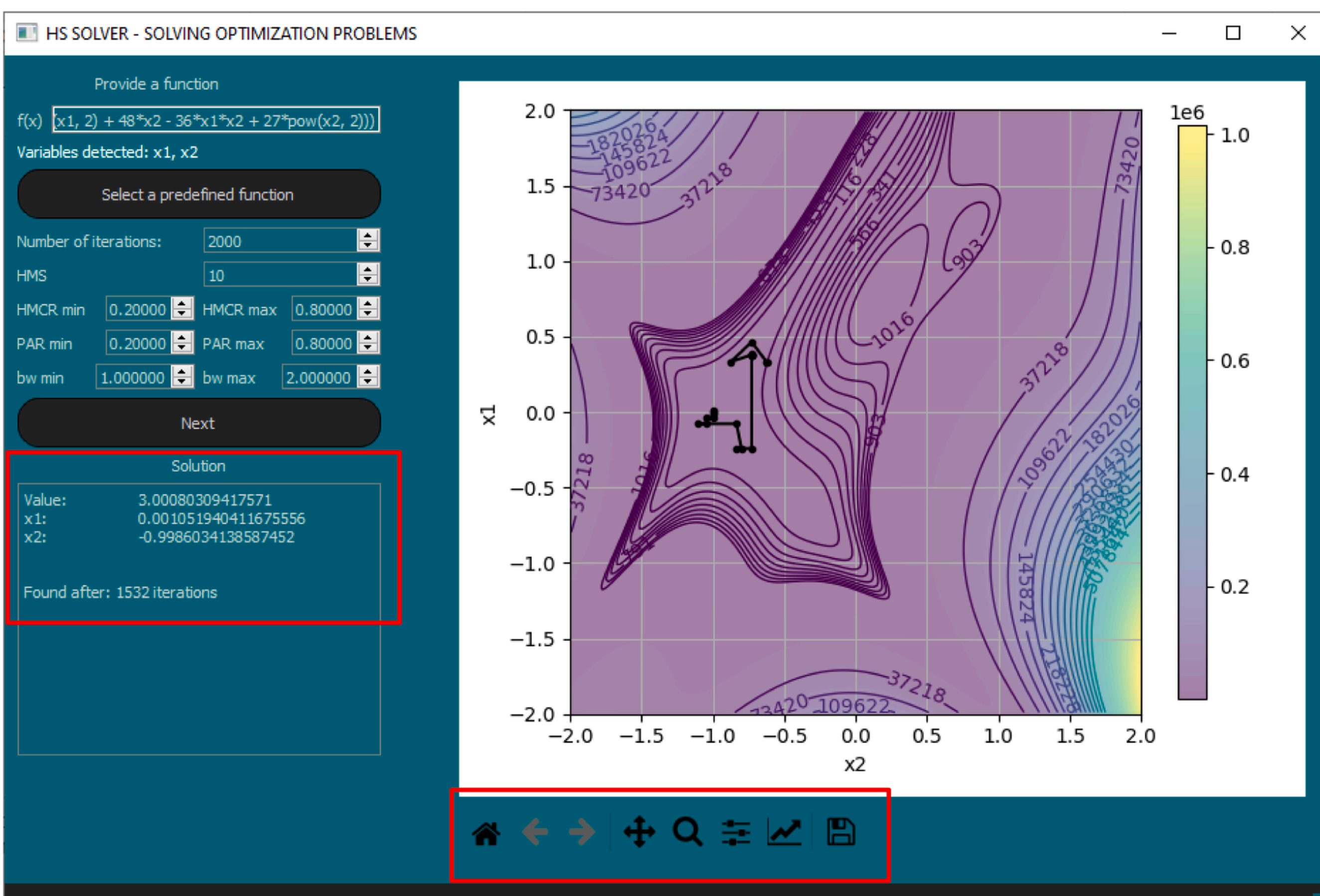

Fig. 4. The function contour plot and the found points.

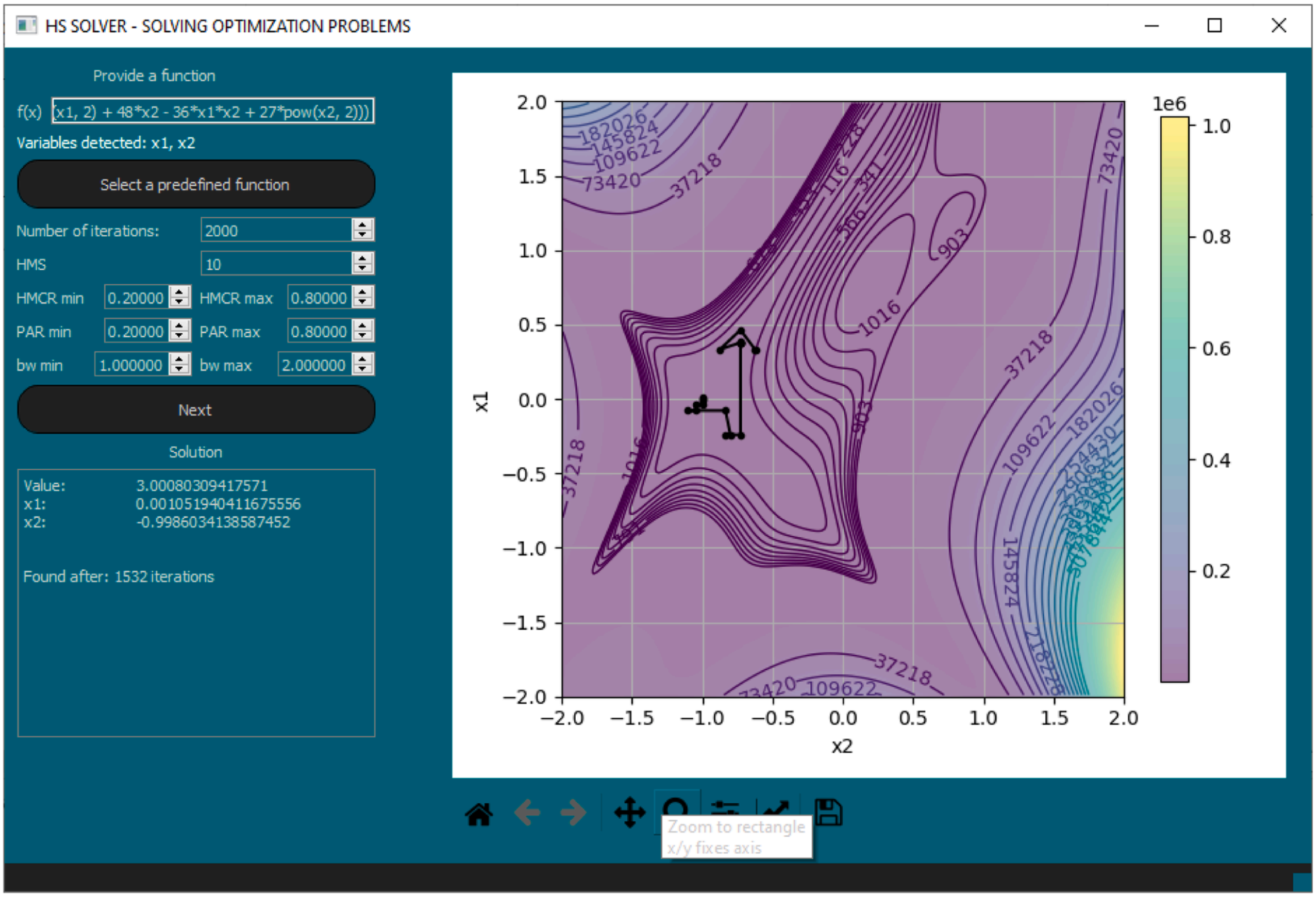

Fig. 5. The global minimum.

engineering, telecommunications, construction, and structural engineering.

In steel engineering, the algorithm proves invaluable in typical structural design optimization problems. Engineers often face the challenge of selecting appropriate steel members for beams and columns to meet stringent serviceability and strength criteria while minimizing material costs. The HS algorithm facilitates the selection process, ensuring that the steel frame achieves minimum weight without compromising on performance standards.

Similarly, the design of Shell and Tube Heat Exchangers (STHX) benefits greatly from the HS algorithm. STHX are prevalent in the process industry due to their simple manufacturing and adaptability to diverse operating conditions. The design process of STHX, which includes thermodynamic and fluid dynamic considerations, cost estimation, and optimization, is inherently complex, drawing on design rules and empirical knowledge from various fields. The HS algorithm streamlines this multifaceted process, enhancing both accuracy and efficiency.

Energy-efficient building design is another area where the HS algorithm excels. As global priorities shift towards improving energy efficiency and environmental performance, new building regulations increasingly emphasize low-emission and energy-efficient designs. The optimal design of residential buildings must balance multiple, often competing, objectives such as optimizing energy consumption, reducing financial costs, and minimizing environmental impact. The HS algorithm adeptly navigates these complexities, providing solutions that align with stringent regulatory standards while achieving optimal performance.

Water distribution network design presents yet another domain where the HS algorithm demonstrates its utility. In this context, the goal is to select optimal pipe diameters to minimize the total cost of the network. Engineers must account for a fixed-pressure supply node and numerous demand nodes, considering heights and distances between nodes (pipe lengths). The HS algorithm efficiently determines the optimal diameter for each pipe segment, ensuring cost-effective and reliable water distribution.

Energy flow optimization is a critical focus in the energy sector, where the objective is to determine the loads in megawatts that must be supplied by specific nodes or buses of the transmission system to minimize costs. The HS algorithm's robust optimization capabilities make it an ideal tool for tackling such challenges, providing precise and cost-effective solutions.

The HS algorithm's application in engineering science underscores its versatility and effectiveness in solving complex optimization problems. Whether in structural engineering, heat exchanger design, energy-efficient building design, water distribution, or energy flow optimization, the HS algorithm consistently delivers high accuracy and efficiency, cementing its role as a critical tool in modern engineering practice.

5. Conclusions

The application developed using the Harmony Search (HS) algorithm demonstrates a robust and efficient approach to solving minimization problems across various domains. By leveraging the improvisational strategies of musicians, the HS algorithm navigates complex solution landscapes with remarkable accuracy and adaptability. This application, with its intuitive interface and comprehensive visualization tools, not only simplifies the optimization process but also enhances the accessibility of advanced computational techniques for practitioners and researchers alike. The successful implementation and performance of the HS algorithm in this software underscore its potential as a valuable tool in engineering, logistics, and beyond, setting a precedent for future developments in optimization technology.

CRediT authorship contribution statement

Fazliddin Makhmudov: Project administration, Methodology. **Dusmurod Kilichev:** Writing – original draft, Software. **Young Im Cho:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

The authors wish to express their profound gratitude to Prof. Zong Woo Geem from the Department of Smart City at Gachon University for his invaluable guidance and instruction regarding the Harmony Search algorithm. We also extend our sincere gratitude to Young Im Cho for their indispensable support and contributions to our research conducted by the Department of Computer Engineering. Their assistance has been instrumental in the successful completion of our project, and we deeply appreciate their dedication and expertise.

References

[1] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8.
[2] Geem ZW. Optimal cost design of water distribution networks using harmony search. Eng Optim 2006;38:259–80.
[3] Li HQ, Li L. A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems. In: International conference on intelligent pervasive computing Korea; 2007.
[4] Lee KS, Geem ZW, Lee SH. The harmony search heuristic for discrete structural optimization. Eng Optim 2005;37:663–84.
[5] Geem ZW. Harmony search applications in industry. Stud Fuzziness Soft Comput 2008;226:117–34.
[6] Geem ZW, Tseng C-L, Park Y. Harmony search for generalized orienteering problem: best touring in China. Lect Notes Comput Sci 2005;3612:741–50.
[7] Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. Comput Struct 2004;82(9–10):781–98.
[8] Li L, Chi SC, Lin G. Genetic algorithm incorporated with harmony procedure and its application to searching of non-circular critical slip surface in soil slopes. J Hydraul Eng 2005;36:91–918.
[9] Geem ZW, Sim KB. Parameter-setting-free harmony search algorithm. Appl Math Comput 2010;217(8):3881–9.